06-05-00

A

PATENT APPLICATION
Express Mail Label No. *EL436467572US*
Attorney Docket No. NA99-08101

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

UTILITY PATENT
APPLICATION TRANSMITTAL LETTER

Asst. Commissioner for Patents
Box Patent Application
Washington, D.C. 20231

Sir:

Enclosed for filing is an [X] original patent application or, [ ] a continuation-in-part patent application, by inventor(s) Igor Muttik, Duncan V. Long, entitled DETECTING COMPUTER VIRUSES OR MALICIOUS SOFTWARE BY PATCHING INSTRUCTIONS INTO AN EMULATOR.

No. of pages in Application: 19; No. of Claims: 36.

No. of Sheets of Drawings: Formal: 2, Informal: 0.

Also enclosed are:

[ ] a claim for foreign priority under 35 U.S.C. §§ 119 and/or 365 in
[ ] a separate document [ ] the declaration;

[ ] a certified copy of the priority document;

[ ] an Associate Power of Attorney;

[ ] ___ verified statement(s) claiming small entity status;

[X] a Combined Declaration and Power of Attorney of the inventors(s);

[ ] a signed Combined Declaration and Power of Attorney of the inventors will follow;

[X] an Assignment document and form PTO-1595;

[ ] a Power of Attorney by Assignee; and

[ ] Information Disclosure Statement and Form PTO-1449.

The fee has been calculated as follows:

CLAIMS					
	NO. OF CLAIMS		EXTRA CLAIMS	RATE	FEE
Basic Application Fee					$690.00
Total Claims	36	MINUS 20 =	16	$18.00=	$288.00
Independent Claims	3	MINUS 3 =	0	$78.00=	$0.00
If multiple dependent claims are presented, add $260.00					0
Total Application Fee					$978.00
If verified statement claiming small entity status is enclosed, subtract 50% of Total Application Fee					
Add Recording Fee of $40.00 if Assignment document is enclosed					$40.00
TOTAL APPLICATION FEE DUE					$1018.00

[X] A check in the amount of $ 1018.00 is enclosed.

[ ] Application fee will follow with missing parts.

[X] Please deduct any underpayments or credit any overpayments to Deposit Account Number 50-1003.

Please direct all correspondence concerning the above-identified application to the following address:

A. Richard Park
Park & Vaughan LLP
508 Second Street, Suite 201
Davis, CA 95616
(530) 759-1661

Respectfully submitted,

By _____
A. Richard Park
Registration No. 41,241

Date: June 1, 2000

"Express Mail" Mailing Label No. **EL436467572US**

PATENT APPLICATION
ATTORNEY DOCKET NO. NA99-08101

DETECTING COMPUTER VIRUSES OR MALICIOUS SOFTWARE BY PATCHING INSTRUCTIONS INTO AN EMULATOR

Inventor(s): Igor Muttik and Duncan V. Long

BACKGROUND

Field of the Invention

The present invention relates to systems for detecting computer viruses and malicious software. More specifically, the present invention relates to a method and an apparatus for emulating computer viruses or other malicious software that operates by patching additional instructions into an emulator in order to aid in the process of detecting, decrypting or disinfecting code containing a computer virus or other malicious software.

Related Art

Malicious software, such as a computer virus, can enter a computer system in a number of ways. It can be introduced on a disk or a CD-ROM that is inserted

into the computer system. It can also enter from a computer network, for example, within an email message.

If malicious software is executed by a computer system, it can cause a number of problems. The software can compromise security, by stealing passwords; by creating a "back door" into the computer system; or by otherwise accessing sensitive information. The software can also cause damage to the computer system, for example, by deleting files or by causing the computer system to fail.

Some types of malicious programs can be easily detected using simple detection techniques, such as scanning for a search string. However, this type of detection process can be easily subverted by converting a malicious algorithm into program code in different ways.

Another approach to detecting malicious software is to run a program on a real machine while attempting to intercept malicious actions. This technique, which is known as "behavior blocking," has a number of disadvantages. In spite of the attempt to intercept malicious actions, the program may nevertheless cause harm to the computer system. Furthermore, the behavior blocking mechanism typically cannot view an entire log of actions in making a blocking determination. Hence, the behavior blocking mechanism may make sub-optimal blocking decisions, which means harmless programs may be blocked or harmful programs may be allowed to execute.

Yet another approach to detecting malicious software is to "emulate" suspect code within an insulated environment in a computer system so that the computer system is protected from malicious actions of the suspect code.

One disadvantage to emulation is that it is almost impossible to provide complete emulation for all program instructions, all operating system calls and operating system environments that may be accessed by a piece of code being

emulated without replicating the entire operating system in the process. Hence, in practice, emulators are typically able to emulate only commonly occurring program instructions and system calls.

This problem can be overcome by updating and recompiling an emulator to implement new system calls and new program instructions as different pieces of malicious software are encountered that make use of these new system calls and new program instructions. However, doing so can lead to logistical problems in keeping emulation programs up to date.

Another problem with current emulators is that they cannot deal with conflicting emulator environments. For example, one virus may be triggered by a system call returning the year 1999, while another virus is triggered by the same system call returning the year 2000.

What is needed is a method and an apparatus for emulating suspect code that can be easily reconfigured to accommodate new program instructions, system calls and emulation environments.

SUMMARY

One embodiment of the present invention provides a system for emulating computer viruses and/or malicious software that operates by patching additional program instructions into an emulator in order to aid in detecting a computer virus and/or malicious software within suspect code. During operation, the system loads a first emulator extension into the emulator. This first emulator extension includes program instructions that aid in the process of emulating the suspect code in order to detect a computer virus and/or malicious software. The system also loads the suspect code into an emulator buffer within a data space of a computer system. Next, the system performs an emulation using the first emulator extension and the suspect code. This emulation is performed within an insulated

environment in the computer system so that the computer system is insulated from malicious actions of the suspect code. During this emulation, the system determines whether the suspect code is likely to exhibit malicious behavior.

In one embodiment of the present invention, loading the first emulator extension into the emulator involves loading the first emulator extension into the emulator buffer within the emulator. In this embodiment, performing the emulation involves emulating the program instructions that comprise the first emulator extension.

In one embodiment of the present invention, emulating the program instructions that comprise the first emulator extension causes the emulator to examine the suspect code looking for patterns that indicate that the suspect code is likely to exhibit malicious behavior.

In one embodiment of the present invention, emulating the program instructions that comprise the first emulator extension causes the program instructions within the first emulator extension to facilitate emulation of the suspect code.

In one embodiment of the present invention, prior to loading the first emulator extension into the emulator buffer, the system emulates the suspect code without using the first emulator extension.

In one embodiment of the present invention, the system additionally loads a second emulator extension into the emulator, and performs a second emulation using the second emulator extension and the suspect code. In a variation on this embodiment, the first emulator extension implements a first emulation environment that conflicts with a second emulation environment that is implemented by the second emulator extension.

In one embodiment of the present invention, loading the first emulator extension involves loading the first emulator extension from a database containing a plurality of different emulator extensions.

In one embodiment of the present invention, the first emulator extension includes code for decrypting an encrypted computer virus.

In one embodiment of the present invention, if a computer virus or other malicious software is detected within the suspect code, the system additionally disinfects the suspect code.

In one embodiment of the present invention, the first emulator extension facilitates emulating a non-standard computer instruction opcode.

In one embodiment of the present invention, the first emulator extension facilitates emulating an uncommonly used operating system call.

BRIEF DESCRIPTION OF THE FIGURES

FIG. 1 illustrates a computer system in accordance with an embodiment of the present invention.

FIG. 2 illustrates the internal structure of an emulator for emulating and analyzing code for malicious behavior in accordance with an embodiment of the present invention.

FIG. 3 is a flow chart illustrating the process of emulating and analyzing code for malicious behavior using emulator extensions in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION

The following description is presented to enable any person skilled in the art to make and use the invention, and is provided in the context of a particular application and its requirements. Various modifications to the disclosed

embodiments will be readily apparent to those skilled in the art, and the general principles defined herein may be applied to other embodiments and applications without departing from the spirit and scope of the present invention. Thus, the present invention is not intended to be limited to the embodiments shown, but is to be accorded the widest scope consistent with the principles and features disclosed herein.

The data structures and code described in this detailed description are typically stored on a computer readable storage medium, which may be any device or medium that can store code and/or data for use by a computer system. This includes, but is not limited to, magnetic and optical storage devices such as disk drives, magnetic tape, CDs (compact discs) and DVDs (digital video discs), and computer instruction signals embodied in a transmission medium (with or without a carrier wave upon which the signals are modulated). For example, the transmission medium may include a communications network, such as the Internet.

Computer System

FIG. 1 illustrates a computer system 106 in accordance with an embodiment of the present invention. Computer system 106 may include any type of computer system, including, but not limited to, a computer system based on a microprocessor, a mainframe computer, a digital signal processor, a personal organizer, a device controller, and a computational engine within an appliance.

Computer system 106 can receive suspect code 108 (which can potentially be malicious) from a number of different sources. Suspect code 108 may be introduced into computer system 106 by a remote host 101 across a network 102. For example, suspect code 108 may be included within an electronic mail (email) message from remote host 101 to computer system 106. Remote host 101 can

include any entity that is capable of sending suspect code 108 across network 102 to computer system 106. Network 102 can include any type of wire or wireless communication channel capable of coupling together computing nodes. This includes, but is not limited to, a local area network, a wide area network, or a combination of networks. In one embodiment of the present invention, network 102 includes the Internet.

Suspect code 108 may additionally be introduced into computer system 106 by encoding suspect code 108 on a computer-readable storage medium, such as disk 104, and introducing disk 104 into computer system 106. Note that disk 104 can generally include any type of computer-readable storage medium, such as a magnetic disk, a magnetic tape and a CD-ROM.

Before executing suspect code 108, computer system 106 uses emulator 110 to analyze suspect code 108. Emulator 110 analyzes suspect code 108 by executing emulator code 203 and emulator extensions 204 as is described below with reference to FIGs. 2 and 3.

Emulator Structure

FIG. 2 illustrates the internal structure of an emulator 110 for emulating and analyzing suspect code 108 for malicious behavior in accordance with an embodiment of the present invention. Emulator 110 includes emulator code 203, emulator buffer 201 and database 206. Emulator code 203 includes code to perform the emulation.

Emulator buffer 201 is a protected region of memory (also known as a sandbox or a working space) in which suspect code 108 is stored and emulated. Emulator buffer 201 stores suspect code 108 as well as emulator extension 204. Emulator buffer 201 and emulator code 203 are designed so that while suspect code 108 that is executing within emulator buffer 201, suspect code 108 cannot

damage or compromise computer system 106. Emulator extension 204 includes additional program instructions that assist emulator code 203 in the emulation process.

Note that emulator buffer 201 is not within the program space of computer system 106, but is instead in the data space. Hence, instructions within emulator extension 204 must themselves be emulated by emulator code 203. In an alternative embodiment of the present invention, emulator extension 204 is loaded as a patch into the program space of computer system 106. In this alternative embodiment, emulator extension can be executed directly on computer system 106.

Emulator extension 204 is retrieved from database 206, which contains a plurality of emulator extensions 208, which can be successively loaded into emulator buffer 201 during the emulation process. Database 206 can include any type of volatile or non-volatile memory or storage device that can be used to store emulator extensions 208. Database 206 can reside within computer system 106, or alternatively, can reside on an external database server that is separate from computer system 106.

During the emulation process, emulator extension 204 can read suspect code 108 looking for patterns indicating the suspect code 108 contains a virus or other type of malicious software. Alternatively, emulator extension 204 can set up an environment that is conducive to emulating suspect code 108. For example, emulator extension 204 can configure the system to emulate uncommonly used system calls or opcodes. This enables emulator code 203 and/or emulator extension 204 to determine of suspect code 108 exhibits malicious behavior. Emulator code 203 (working with emulator extension 204) ultimately outputs a decision 212 indicating whether suspect code 108 is malicious or not.

Note that emulator extension 204 can be emulated in a number of different ways. (1) Emulator extension 204 can be emulated as part of suspect code 108 by patching the emulator extension 204 into suspect code 108, possibly replacing, overlapping or overwriting portions of suspect code 108. In this case, the location where the patching occurs is defined in the database 206. (2) Emulator extension 204 can be executed before the suspect code 108 is executed, which enables emulator extension 204 to set up the environment that emulator extension 204 is responsible for handling. After this environment is set up, emulator extension 204 passes control suspect code 108. (3) Emulator extension 204 can replace suspect code 108 entirely. In this case, the suspect code 108 is not emulated at all, and emulator extension 204 produces decision 212 after analyzing the suspect code 108 as data. (4) Emulator extension 204 can be emulated after the suspect code 108 is emulated. This allows emulator extension 204 to analyze the results of running the suspect code 108 in order to produce decision 212.

Process of Emulation

FIG. 3 is a flow chart illustrating the process of emulating and analyzing code for malicious behavior using emulator extensions in accordance with an embodiment of the present invention. The system starts by receiving suspect code 108 from one of a number of possible sources as is described above with reference to FIG. 1 (step 302). The system loads this suspect code into emulator buffer 201 (step 304).

Next, the system runs emulator 110 (step 306). This causes suspect code 108 to be examined and/or emulated by emulator code 203. During the emulation process, the system determines whether or not suspect code 108 contains code that is likely to exhibit malicious behavior (step 308). If so, the system reports the malicious code to a system user or system administrator (step 310).

If no malicious code is detected, the system determines if there are any emulator extensions remaining in database 206 that have not already been used (step 312). If not, the system proceeds to the next file containing suspect code to repeat the entire process (step 314).

Otherwise, if there are emulator extensions remaining, the system loads the next emulator extension into emulator 110 (step 315). In one embodiment of the present invention, this involves loading emulator extension 204 into emulator buffer 201 within emulator 110. In an alternative embodiment, this involves loading emulator extension 204 into the program space of computer system 106 so that it can work in concert with emulator code 203 in performing a subsequent emulation.

Next, the system sets up emulator 110 to run emulator extension 204 (step 316). This may involve configuring emulator code 203 to initially run emulator extension 204. Next, the system returns to step 306 to continue with the emulation process using the new emulator extension.

Note that by using multiple emulator extensions it is possible to deal with conflicting emulator environments. For example, a first emulator extension can configure emulator 110 to detect a virus that is triggered by a system call returning the year 1999, while a second emulator extension can configure emulator 110 to detect a virus that is triggered by the same system call returning the year 2000.

The foregoing descriptions of embodiments of the invention have been presented for purposes of illustration and description only. They are not intended to be exhaustive or to limit the present invention to the forms disclosed. Accordingly, many modifications and variations will be apparent to practitioners skilled in the art. Additionally, the above disclosure is not intended to limit the present invention. The scope of the present invention is defined by the appended claims.

What Is Claimed Is:

1. A method for emulating computer viruses and/or malicious software that operates by patching additional program instructions into an emulator in order to aid in detecting a computer virus and/or malicious software within suspect code, the method comprising:

receiving the suspect code;

loading the suspect code into an emulator buffer within a data space of a computer system;

loading a first emulator extension into the emulator, the first emulator extension including program instructions that aid in the process of emulating the suspect code in order to detect a computer virus and/or malicious software;

performing an emulation using the first emulator extension and the suspect code, the emulation being performed within an insulated environment in the computer system so that the computer system is insulated from malicious actions of the suspect code; and

determining whether the suspect code is likely to exhibit malicious behavior based upon the emulation.

2. The method of claim 1, wherein loading the first emulator extension into the emulator includes loading the first emulator extension into the emulator buffer within the emulator; and

wherein performing the emulation includes emulating the program instructions that comprise the first emulator extension.

3. The method of claim 2, wherein emulating the program instructions that comprise the first emulator extension causes the emulator to

examine the suspect code looking for patterns that indicate that the suspect code is likely to exhibit malicious behavior.

4. The method of claim 2, wherein emulating the program instructions that comprise the first emulator extension causes the program instructions within the first emulator extension to facilitate emulation of the suspect code.

5. The method of claim 1, further comprising emulating the suspect code prior to loading the first emulator extension into the emulator buffer.

6. The method of claim 1, further comprising:

loading a second emulator extension into the emulator; and

performing a second emulation using the second emulator extension and the suspect code.

7. The method of claim 6, wherein the first emulator extension and the second emulator extension provide support for conflicting emulator environments.

8. The method of claim 1, wherein loading the first emulator extension involves loading the first emulator extension from a database containing a plurality of different emulator extensions.

9. The method of claim 1, wherein the first emulator extension includes code for decrypting an encrypted computer virus and other encrypted malicious code.

10. The method of claim 1, further comprising if a computer virus or other malicious software is detected within the suspect code, disinfecting the suspect code.

11. The method of claim 1, wherein the first emulator extension facilitates emulating a non-standard computer instruction opcode.

12. The method of claim 1, wherein the first emulator extension facilitates emulating an uncommonly used operating system call.

13. A computer-readable storage medium storing instructions that when executed by a computer cause the computer to perform a method for emulating computer viruses and/or malicious software that operates by patching additional program instructions into an emulator in order to aid in detecting a computer virus and/or malicious software within suspect code, the method comprising:

receiving the suspect code;

loading the suspect code into an emulator buffer within a data space of a computer system;

loading a first emulator extension into the emulator, the first emulator extension including program instructions that aid in the process of emulating the suspect code in order to detect a computer virus and/or malicious software;

performing an emulation using the first emulator extension and the suspect code, the emulation being performed within an insulated environment in the computer system so that the computer system is insulated from malicious actions of the suspect code; and

determining whether the suspect code is likely to exhibit malicious behavior based upon the emulation.

14. The computer-readable storage medium of claim 13, wherein loading the first emulator extension into the emulator includes loading the first emulator extension into the emulator buffer within the emulator; and

wherein performing the emulation includes emulating the program instructions that comprise the first emulator extension.

15. The computer-readable storage medium of claim 14, wherein emulating the program instructions that comprise the first emulator extension causes the emulator to examine the suspect code looking for patterns that indicate that the suspect code is likely to exhibit malicious behavior.

16. The computer-readable storage medium of claim 14, wherein emulating the program instructions that comprise the first emulator extension causes the program instructions within the first emulator extension to facilitate emulation of the suspect code.

17. The computer-readable storage medium of claim 13, wherein the method further comprises emulating the suspect code prior to loading the first emulator extension into the emulator buffer.

18. The computer-readable storage medium of claim 13, wherein the method further comprises:

loading a second emulator extension into the emulator; and

performing a second emulation using the second emulator extension and the suspect code.

19. The computer-readable storage medium of claim 18, wherein the first emulator extension and the second emulator extension provide support for conflicting emulator environments.

20. The computer-readable storage medium of claim 13, wherein loading the first emulator extension involves loading the first emulator extension from a database containing a plurality of different emulator extensions.

21. The computer-readable storage medium of claim 13, wherein the first emulator extension includes code for decrypting an encrypted computer virus and other encrypted malicious code.

22. The computer-readable storage medium of claim 13, wherein if a computer virus or other malicious software is detected within the suspect code, the method further comprises disinfecting the suspect code.

23. The computer-readable storage medium of claim 13, wherein the first emulator extension facilitates emulating a non-standard computer instruction opcode.

24. The computer-readable storage medium of claim 13, wherein the first emulator extension facilitates emulating an uncommonly used operating system call.

25. An apparatus that emulates computer viruses and/or malicious software that operates by patching additional program instructions into an emulator in order to aid in detecting a computer virus and/or malicious software within suspect code, the apparatus comprising:

a loading mechanism that is configured to load the suspect code into an emulator buffer within a data space of a computer system;

wherein the loading mechanism is additionally configured to load a first emulator extension into the emulator, the first emulator extension including program instructions that aid in the process of emulating the suspect code in order to detect a computer virus and/or malicious software;

an emulation mechanism that is configured to perform an emulation using the first emulator extension and the suspect code, the emulation being performed within an insulated environment in the computer system so that the computer system is insulated from malicious actions of the suspect code; and

a determination mechanism that is configured to determine whether the suspect code is likely to exhibit malicious behavior based upon the emulation.

26. The apparatus of claim 25, wherein the loading mechanism is configured to load the first emulator extension into the emulator buffer within the emulator; and

wherein the emulation mechanism is configured to emulate the program instructions that comprise the first emulator extension.

27. The apparatus of claim 26, wherein emulating the program instructions that comprise the first emulator extension causes the emulation mechanism to examine the suspect code looking for patterns that indicate that the suspect code is likely to exhibit malicious behavior.

28. The apparatus of claim 26, wherein emulating the program instructions that comprise the first emulator extension causes the emulation mechanism to facilitate emulation of the suspect code.

29. The apparatus of claim 25, wherein the emulator is configured to emulate the suspect code prior to loading the first emulator extension into the emulator buffer.

30. The apparatus of claim 25, wherein the loading mechanism is additionally configured to:

load a second emulator extension into the emulator; and to

perform a second emulation using the second emulator extension and the suspect code.

31. The apparatus of claim 30, wherein the first emulator extension and the second emulator extension provide support for conflicting emulator environments.

32. The apparatus of claim 25, wherein the loading mechanism is configured to load the first emulator extension from a database containing a plurality of different emulator extensions.

33. The apparatus of claim 25, wherein the first emulator extension includes code for decrypting an encrypted computer virus and other encrypted malicious code.

34. The apparatus of claim 25, further comprising a disinfecting mechanism that is configured to disinfect the suspect code if a computer virus or other malicious software is detected within the suspect code.

35. The apparatus of claim 25, wherein the first emulator extension is configured to facilitate emulating a non-standard computer instruction opcode.

36. The apparatus of claim 25, wherein the first emulator extension is configured to facilitate emulating an uncommonly used operating system call.

DETECTING COMPUTER VIRUSES OR MALICIOUS SOFTWARE BY PATCHING INSTRUCTIONS INTO AN EMULATOR

ABSTRACT

One embodiment of the present invention provides a system for emulating computer viruses and/or malicious software that operates by patching additional program instructions into an emulator in order to aid in detecting a computer virus and/or malicious software within suspect code. During operation, the system loads a first emulator extension into the emulator. This first emulator extension includes program instructions that aid in the process of emulating the suspect code in order to detect a computer virus and/or malicious software. The system also loads the suspect code into an emulator buffer. Next, the system performs an emulation using the first emulator extension and the suspect code. This emulation is performed within an insulated environment in a computer system so that the computer system is insulated from malicious actions of the suspect code. During this emulation, the system determines whether the suspect code is likely to exhibit malicious behavior. In one embodiment of the present invention, loading the first emulator extension into the emulator involves loading the first emulator extension into the emulator buffer within the emulator. In this embodiment, performing the emulation involves emulating the program instructions that comprise the first emulator extension.

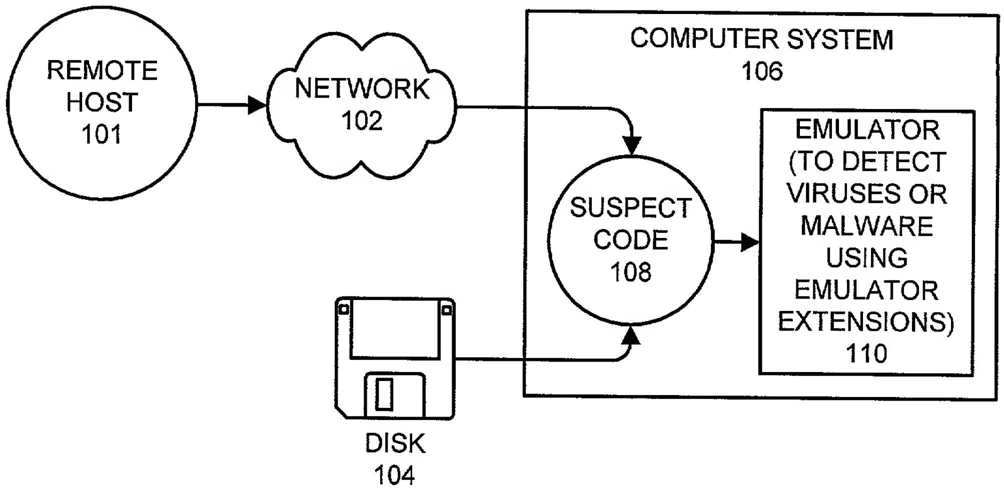
REMOTE HOST 101
NETWORK 102
COMPUTER SYSTEM 106
SUSPECT CODE 108
EMULATOR (TO DETECT VIRUSES OR MALWARE USING EMULATOR EXTENSIONS) 110
DISK 104

FIG. 1

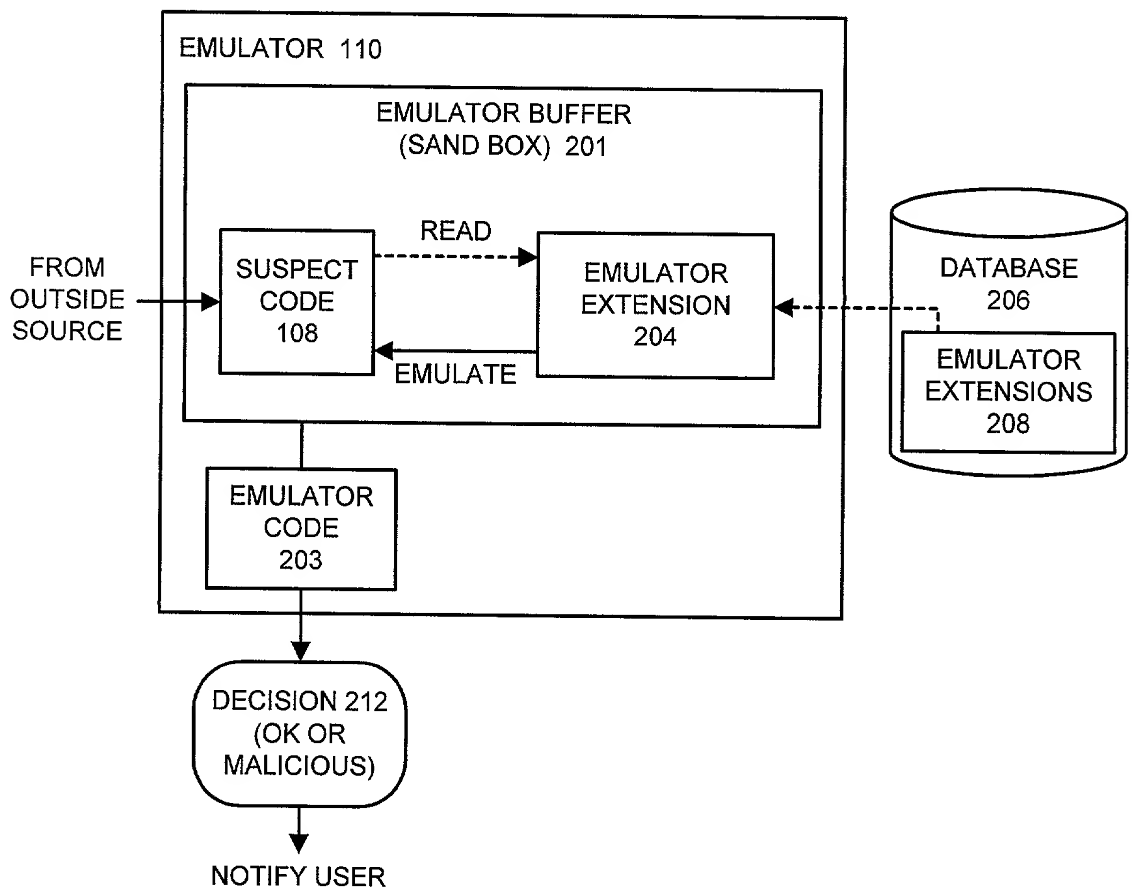
EMULATOR 110
EMULATOR BUFFER (SAND BOX) 201
FROM OUTSIDE SOURCE
SUSPECT CODE 108
READ
EMULATE
EMULATOR EXTENSION 204
DATABASE 206
EMULATOR EXTENSIONS 208
EMULATOR CODE 203
DECISION 212 (OK OR MALICIOUS)
NOTIFY USER

FIG. 2

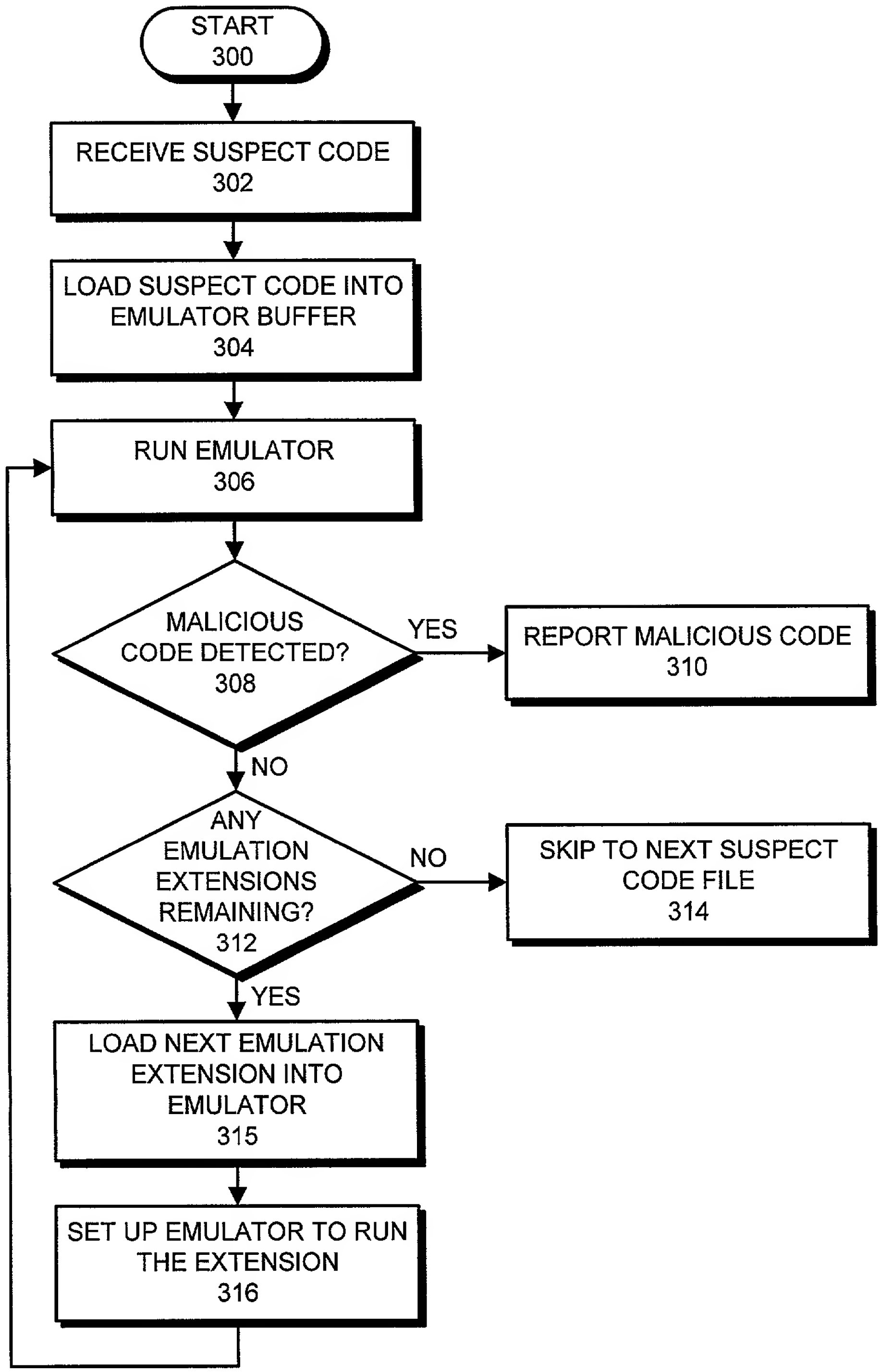

FIG. 3

Attorney Docket No. NA99-08101

COMBINED DECLARATION AND POWER OF ATTORNEY

As a below-named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below by my name;

I believe I am the original, first and sole inventor, if only one name is listed below, or an original, first and joint inventor if multiple names are listed below, of the subject matter which is claimed and for which a patent is sought on the invention entitled:

DETECTING COMPUTER VIRUSES OR MALICIOUS SOFTWARE BY PATCHING INSTRUCTIONS INTO AN EMULATOR

for which a patent application:

☒ is attached hereto.

☐ was filed in the United States on _ as Application No. _;
☐ with amendment(s) filed on _____ *(if applicable)*.

I hereby state that I have reviewed and understand the contents of the application identified above, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information known to me to be material to the examination of this application in accordance with Title 37, Code of Federal Regulations, §1.56, which states in relevant part:

> Each individual associated with the filing and prosecution of a patent application has a duty of candor and good faith in dealing with the Office, which includes a duty to disclose to the Office all information known to that individual to be material to patentability as defined in this section . . The duty to disclose all information known to be material to patentability is deemed to be satisfied if all information known to be material to patentability of any claim issued in a patent was cited by the Office or submitted to the Office....

I hereby claim foreign priority benefits under Title 35, United States Code, §119(a)-(d), of any foreign application(s) for patent or inventor's certificate as indicated below and have also identified below any foreign application for patent or inventor's certificate on this invention having a filing date before that of the application on which priority is claimed:

EARLIEST FOREIGN APPLICATION(S), IF ANY, FILED PRIOR TO THE FILING DATE OF THE APPLICATION			
APPLICATION NUMBER	COUNTRY	DATE OF FILING (Day, Month, Year)	PRIORITY CLAIMED
			YES ☐ NO ☐

I hereby claim the benefit under Title 35, United States Code, §119(e), of any United States provisional application(s) listed below:

APPLICATION NUMBER	DATE OF FILING

I hereby claim the benefit under Title 35, United States Code, §120, of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §112, I acknowledge the duty to disclose information that is material to patentability as defined in Title 37, Code of Federal Regulations, §1.56, which became available between the filing date of the prior application and the national or PCT international filing date of this application:

APPLICATION NUMBER	DATE OF FILING	STATUS		
		PATENTED	PENDING	ABANDONED

I hereby appoint Daniel E. Vaughan (Reg. No 42,199) and A. Richard Park (Reg. No. 41,241) to prosecute this application

Attorney Docket No. NA99-08101

and transact all business in the Patent and Trademark Office connected therewith, and to file, prosecute and transact all business in connection with international applications directed to said invention.

Address correspondence to:
Park & Vaughan LLP
508 Second Street, Suite 201
Davis, CA 95616

22835

PATENT TRADEMARK OFFICE

Direct telephone calls to:
A. Richard Park
(530) 759-1661

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Title 18, United States Code, §1001, and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

1	Name and Citizenship	Igor Muttik	United Kingdom
	Residence Address	Woodlands, Kingsdale Road, Berkhamsted, Herts, HP4 3BS, England	
	Postal Address *(if different from Residence)*		
	Signature and Date		Date 1 JUNE 2000
2	Name and Citizenship	Duncan V. Long	United Kingdom
	Residence Address	128 Western Road, Tring, Hertfordshire, HP23 4BJ, England	
	Postal Address *(if different from Residence)*		
	Signature and Date	Duncan V	Date 01. JUNE. 2000
3	Name and Citizenship		
	Residence Address		
	Postal Address *(if different from Residence)*		
	Signature and Date		Date
4	Name and Citizenship		
	Residence Address		
	Postal Address *(if different from Residence)*		
	Signature and Date		Date
5	Name and Citizenship		
	Residence Address		
	Postal Address *(if different from Residence)*		
	Signature and Date		Date

Additional inventor name(s) and signature(s) attached?: YES ☐ NO ☒